

CLAIMS

What is claimed is:

- 004400 0050108 0050560
1. A computer with an operating system and persistent memory, comprising:
a memory, comprising:
a non-persistent memory region, directly accessible by the operating system; and
a persistent memory region; and
an intermediary program in communication with the operating system and the persistent memory region,
wherein the intermediary program enables the operating system to address a persistent memory region.
2. The computer of claim 1 wherein a non-persistent memory region and a persistent memory region are different physical memories.
3. The computer of claim 1 wherein the intermediary program is a device driver.
4. The computer of claim 1 additionally comprising:
a basic input/output system (BIOS), which prevents direct access to the persistent memory region by the operating system.
5. The computer of claim 1 wherein the persistent memory region is allocated to redundant CPU memory locations.
6. The computer of claim 1 additionally comprising:
a non-volatile memory,
wherein the non-volatile memory contains information concerning the configuration of the persistent memory region.
7. The computer of claim 1 additionally comprising:

2 a non-volatile memory,
3 wherein the non-volatile memory contains information concerning the
4 configuration of the non-persistent memory region.

1 8. The computer of claim 1 additionally comprising:
2 a file containing system settings,
3 wherein the file containing system settings contains information concerning the
4 configuration of the persistent memory region.

1 9. The computer of claim 1 additionally comprising:
2 a file containing system settings,
3 wherein the file containing system settings contains information concerning the
4 configuration of the non-persistent memory region.

1 10. The computer of claim 1 wherein the persistent memory region comprises:
2 a look-aside buffer, comprising:
3 a set of state bits, and
4 a buffer region for the storage of data,
5 wherein the look-aside buffer is used for the atomic storage and update of write
6 requests.

1 11. A storage medium with an encoded program which when loaded into a computer
2 having an operating system and a memory partitioned into a non-persistent memory
3 region and a persistent memory region, provides the computer with a persistent memory,
4 said program comprising the steps of:

5 (a) reading from the persistent memory region in response to requests coming
6 from the operating system; and

7 (b) writing to the persistent memory region in response to requests coming from
8 the operating system.

- 1 12. The encoded program of claim 11 wherein the persistent memory region of the
2 computer comprises a look-aside buffer, itself comprising a set of state bits and a buffer
3 region for the storage of data, and step (b) further comprises the steps of:
- 4 (b-a) setting the state bits to a first value before writing the contents of a request
5 to the buffer region;
- 6 (b-b) writing the contents of a request to the buffer region;
- 7 (b-c) setting the state bits to a second value after successful completion of the
8 writing of the contents of a request to the buffer region;
- 9 (b-d) copying the contents of the buffer region to the appropriate location in the
10 persistent memory; and
- 11 (b-e) setting the state bits to a third value after successfully copying the contents
12 of the buffer region to the appropriate location in the persistent memory.
- 1 13. In a computer system comprising an operating system, an intermediary program,
2 and a memory, a method for providing persistent memory, the method comprising the
3 steps of:
- 4 (a) partitioning the memory into a non-persistent memory region and a persistent
5 memory region;
- 6 (b) providing an intermediary program in communication with the persistent
7 memory region such that the persistent memory region is accessible to the operating
8 system solely through the device driver.
- 1 14. The method of claim 13 wherein the contents of the persistent memory region
2 retain their integrity during a boot cycle.
- 1 15. The method of claim 13 wherein the persistent memory region comprises a look-
2 aside buffer, which the device driver uses for the atomic update and storage of write
3 requests.
- 1 16. The method of claim 13 wherein step (a) comprises the steps of:

- 2 (a-a) reading a stored address defining the start address of the persistent
3 memory region;
- 4 (a-b) reading a stored value defining the size of the persistent memory region;
5 and
- 6 (a-c) creating a persistent memory region at the start address equal in size to the
7 stored value defining the size of the persistent memory region.
- 1 17. The method of claim 16 wherein the program reads the stored addresses and
2 stored values defining the size of the persistent memory region from non-volatile
3 memory.
- 1 18. An operating system memory environment comprising:
2 a first memory mode region accessible to users and to the operating system;
3 a second memory mode region accessible only to the operating system and not to
4 users; and
5 a third memory mode region not accessible by users and not directly accessible by
6 the operating system.
- 1 19. The operating system memory environment of claim 18 wherein the operating
2 system is a Microsoft Windows operating system.
- 1 20. A computer with a memory-mapped operating system and persistent memory,
2 comprising:
3 a volatile memory, comprising:
4 non-persistent memory region, directly accessible by the operating system;
5 and
6 a persistent memory region, whose locations are not mapped by the
7 operating system.
- 1 21. The computer of claim 20 further comprising:

2 a device driver in communication with the operating system and the persistent
3 memory region,
4 wherein the operating system addresses the persistent memory region via the
5 device driver.

1 22. The computer of claim 21 wherein the persistent memory region comprises a
2 look-aside buffer, which the device driver uses for the atomic update and storage of write
3 requests.

1 23. The computer of claim 20 wherein the locations of the persistent memory region
2 are mapped by the operating system or a user application once the boot cycle is complete.